Langages à balises : une introduction

Yannick Prié
UFR Informatique – Université Lyon 1

UE2.2 – Master SIB M1 – 2008-2009

Objectif généraux du cours

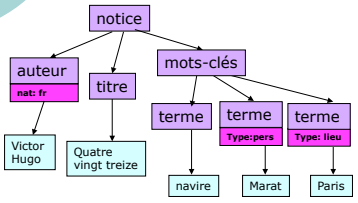
- Comprendre les grands principes de la représentation de données et de documents numériques à l'aide d'un langage à balises.
- Découvrir XML, son histoire et son fonctionnement
- Définir des langages basés sur XML à l'aide de DTD
- S'initier à la transformation de documents en utilisant XSL et un moteur XSLT
- Apprendre les bases de XHTML pour la génération de pages web

Objectifs de ce cours introductif

- Introduction aux langages à balise et à leurs principes
  - arbres
  - grammaires
  - langages à balises
- Histoire de ces langages
- Présentation de la galaxie XML et de la suite du cours

Idée générale

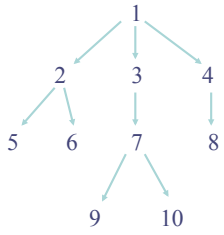
- Représenter de l'information dans des structures arborescentes
- Coder ces structures dans des fichiers, qui pourront être échangés



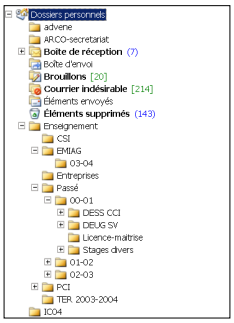

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<notice>
<auteur nat="fr">Victor Hugo
</auteur>
<titre>Quatre vingt treize</titre>
<mots-clés>
<terme>navire</terme>
<terme Type="pers">Marat</terme>
<terme Type="lieu">Paris</terme>
</mots-clés>
</notice>
```

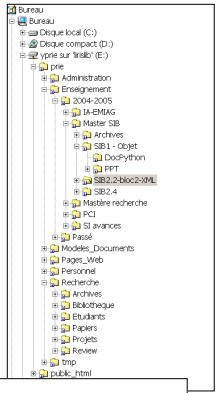

Parler des arbres

- Arbre
- Noeud
  - nœuds fils et pères
- Racine
- Feuille
- Chemin
  - suite de nœud
- Branche
  - chemin se terminant sur une feuille
- Ancêtres et descendants
- Taille d'un arbre
  - nombre de nœuds
- Profondeur d'un nœud



Les arbres sont partout !

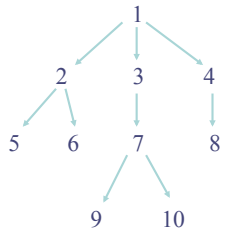




chemin
E:\prie\Enseignement\2004-2005\
Master SIB\SIB2.2-bloc2-XML

Parcours d'arbre

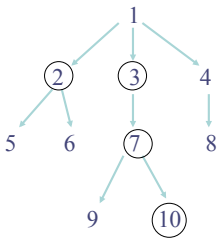
- Largeur d'abord
  1
  → 2 → 3 → 4
  → 5 → 6 → 7 → 8
  → 9 → 10
- Profondeur d'abord
  1 → 2 → 5 → 6
  → 3 → 7 → 9 → 10
  → 4 → 8



Algorithme de parcours

- Objectif :
  - compter les nœuds entourés

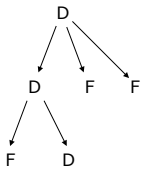
```
/* Fonction pour comptage local */
compter_localement (nœud)
    Si il y a un cercle
    Alors n ← n +1
    Pour tous les nœuds fils n_i,
        Compter_localement (n_i)
/* Appel général */
n ← 0
compter_localement (nœud 1)
afficher n
```



- Remarques
  - parcours en profondeur d'abord
  - autant de comptages locaux que de nœuds
  - marche sur n'importe quel arbre : on part de la racine et on parcours tout
  - pas de vision globale de l'arbre

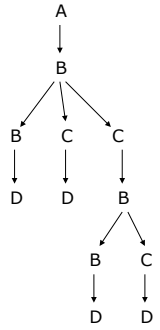
Notion de grammaire

- Système formel
  - vocabulaire + règles de production
  - permet de définir un arbre
- Exemple
  - vocabulaire
    - D (Dossier)
    - F (fichier)
  - Règle
    - Départ : D
    - D → (D|F)*
    - Avec
      - * == zéro ou plus
      - | == ou



Autre exemple

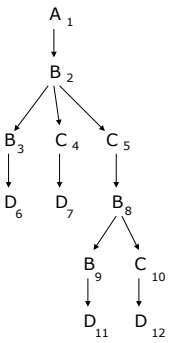
- Vocabulaire
  - A, B, C, D
- Règles
  - Départ : A
  - A → B+
  - avec
    - + == 1 ou plus
  - B → BC* | D
  - C → (D | B)
- Question
  - quel est l'arbre le plus petit que l'on peut écrire avec cette grammaire ?



Arbres et séquences d'octets

- Fichier
  - suite d'octets
- Objectif
  - représenter un arbre dans un fichier
- Solution
  - décrire l'arbre comme un ensemble d'éléments qui se contiennent les uns les autres.
  - représenter les éléments entre deux balises
    - balises ouvrantes
      - on les notera par exemple `<nom>`
    - balises fermantes
      - on les notera par exemple `</nom>`

Arbres et séquences



Eléments

$A_1 \subset B_2$
$B_2 \subset B_3\ C_4\ C_5$
$B_3 \subset D_6$
$C_4 \subset D_7$
$C_5 \subset B_8$
$B_8 \subset B_9\ C_{10}$
$B_9 \subset D_{11}$
$C_{10} \subset D_{12}$

Eléments et balises

```
<A>
  <B>
    <B>
      <D></D>
    </B>
    <C>
      <D></D>
    </C>
    <C>
      <B>
        <B><D></D></B>
        <C><D></D></C>
      </B>
    </C>
  </B>
</A>
```

Langages à balises

- Tous les langages ayant pour objectif de représenter de l'information en utilisant des balises
- Définis par
  - vocabulaire
    - noms des éléments
  - grammaire
    - mode d'organisation des éléments
      - des éléments en contiennent d'autres
  - + attributs des éléments
    - un peu plus de structure (voir cours XML)
- Une description
  - ensemble d'éléments organisés dans un fichier
  - contenus terminaux (texte)

Familles de langages à balises

- Décrire une notice bibliographique
  - notice, titre, auteur, mots-clés, terme, résumé, ...
- Décrire un poème :
  - poeme, quatrain, tercet, vers, ...

```
<poeme type="sonnet">
<quatrain>
<vers>Je vis, je meurs ; je me brûle
et me noie.</vers>
<vers>J'ai chaud extrême en
endurant froidure ; </vers>
<vers> ... </vers>
<vers> ... </vers>
</quatrain>
...
</poeme>
```

```
<notice>
<auteur nat="fr">Victor Hugo
</auteur>
<titre>Quatre vingt treize</titre>
<mots-clés>
<terme>navire</terme>
<terme Type="pers">Marat</terme>
<terme Type="lieu">Paris</terme>
</mots-clés>
</notice>
```

→ vocabulaires différents
→ grammaires différentes
→ mais *même manière d'exprimer les descriptions*

Notion de métalangage

- Langage avec lequel on peut définir d'autres langages
- Pour les langages à balises
  - langage exprimant la manière dont on peut décrire une famille de langages à balise
    - comment exprimer les éléments ?
    - comment organiser les éléments ?
- Exemples de métalangages
  - SGML
    - permet de définir : TEI, HTML, ...
  - XML
    - permet de définir : SVG, TEI, XHTML, ...

Dans la suite

- Petite histoire des langages à balises et des applications liées
  - de SGML à XML

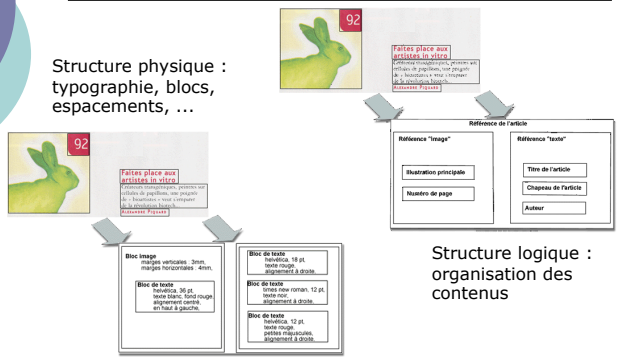
Représentation de documents

- Document numérique
  - manipulations et gestion par des ordinateurs
- Document structuré
  - séparation de la structure physique et de la structure logique
    - séparation forme / contenu
- D'où possibilité
  - de manipuler la structure logique des documents
  - d'accéder au texte des différentes parties logiques des documents
  - de générer plusieurs structures physiques à partir d'une structure logique

(d'après http://sophia.univ-lyon2.fr/didacticiel/unite1/module2.html)

Structures logique / physique

Structure physique : typographie, blocs, espacements, ...



Structure logique : organisation des contenus

Balisage de texte

- Idée
  - marquer des zones des textes pour les qualifier
    - les balises ouvrantes et fermantes délimitent les éléments de description
    - la structure logique est un arbre « ajouté » au texte

```
<p>Il est de tradition de présenter un langage de programmation à l'aide d'un premier exemple comme : <eg> CHAR*20 GRTG GRTG = 'BONJOUR TOUT LE MONDE' PRINT *, GRTG END </eg></p>
<p>Dans cet exemple, on commence par déclarer la variable <ident>GRTG</ident>, dans la ligne <kw>CHAR*20 GRTG</kw>, qui identifie <ident>GRTG</ident> comme formée de 20 octets de type <kw>CHAR</kw>. On affecte alors à cette variable la valeur <mentioned>BONJOUR TOUT LE MONDE</mentioned>. Suivent alors l'ordre d'impression <kw>PRINT</kw> et l'instruction finale <kw>END</kw>.</p>
```

p : servira à la mise en page

eg, kw,mentioned : seront mis en évidence dans la structure physique

kw,mentioned : utilisés pour construire un index

etc.

SGML

- Objectif : représenter l'information contenue dans un document indépendamment
  - des systèmes utilisés pour la saisie et le traitement
  - de la forme physique qu'il sera amené à prendre (papier, CD-ROM, web...)
  - des langues et des alphabets, latins ou non
  - des applications
- Naissance chez IBM (années soixante)
  - GML
  - gestion de la documentation technique
- Normalisation 1986 ISO-8879
  - une dizaine d'années de travail
- Utilisation
  - Description des documents dans les grosses organisations
    - complexité des langages
    - lourdeur et cherté des outils (chaîne de traitement)
    - Journal Officiel, grosses entreprises/documenations, éditeurs...
  - Echange des documents

SGML : principes

- Métalangage
  - permet de décrire des modèles (grammaires)
- Notion de DTD
  - Document Type Definition
  - Permet de décrire un modèle
    - un type de document
- Un document SGML
  - Est une instance du type de document
  - Doit être conforme à la DTD associée

(d'après http://www.cavi.univ-paris3.fr/ilpga/ilpga/tal/)

SGML : exemple

Instance

```
<!DOCTYPE memo SYSTEM "memo.dtd">
<memo statut="conf">
<auteur>Serge Fleury</auteur>
<dest>
<nom>André Salem</nom>
<nom>Pollet Samvelian</nom>
</dest>
<sujet>Cours SLFE6</sujet>
<corps>
<par>Veuillez noter que le cours SLFE6 sur les documents électronique aura bel et bien lieu au mois de mai 2002</par>
<par>S'il y avait des changements de votre côté, veuillez m'en aviser dans les plus brefs délais.</par>
</corps>
</memo>
```

DTD (memo.dtd)

```
<!-- DTD utilisable pour baliser les memos en SGML -->
<!ELEMENT memo -- ((auteur & (date?) & sujet & dest & (cc?)), corps)>
<!ATTLIST memo statut (conf | pub) pub>
<!ELEMENT (dest | cc) -- (nom+)>
<!ELEMENT corps -- (par*)>
<!ELEMENT (auteur | date | sujet | nom | par) -- (#PCDATA)>
```

Un élément corps contient un nombre quelconque de paragraphes

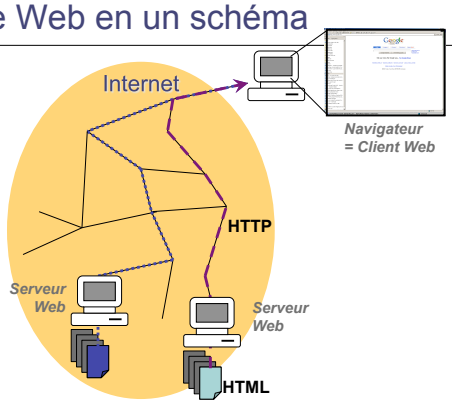
Un élément dest ou cc contient au moins un nom

HTML

- 1991 – CERN – Tim Berners Lee
- Basé sur
  - Principes de l'hypertexte
  - Client/serveur sur IP
- Principes
  - Des serveurs peuvent fournir des documents hypertextes
  - Les documents seront décrits en suivant une DTD SGML → HTML (HyperText Markup Language)
  - Les liens sont décrits avec leur cible (URL)
  - Un client (navigateur)
    - permet de lire (présenter) les documents HTML
    - charge un nouveau document quand on active un lien
  - Protocole d'échange : HTTP (HyperText Transfert Protocol)

(d'après cours PCI – Web http://pci.univ-lyon1.fr)

Le Web en un schéma



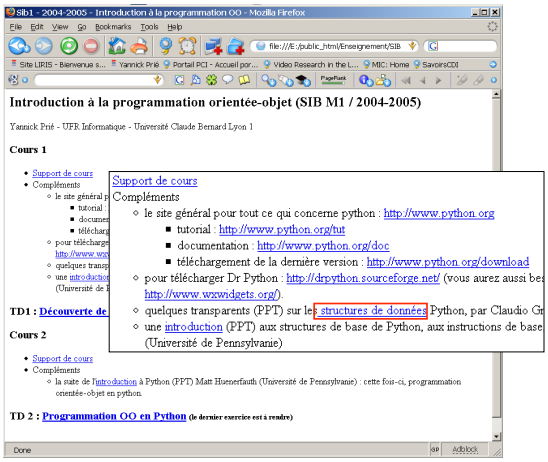
HTML : notion d'URL

- Uniform Ressource Locator
  - permet d'identifier une ressource sur le réseau
- Une ressource peut être
  - une page Web
  - une image (seule ou utilisée dans une page Web)
  - un programme
  - un fichier à télécharger…
- Une URL indique
  - un protocole (langage de communication entre deux programmes sur deux machines)
    - FTP (File Transfert Protocol),
    - HTTP (HyperText Transfert Protocol)…
  - l'adresse d'un serveur
  - un chemin dans l'arborescence des fichiers
- Forme générale : `protocole://adresse/chemin`
  - Exemples
    `http://www.univ-lyon1.fr/`
    `http://www710.univ-lyon1.fr/~yprie/Enseignement/SIB/SIB-UE3-bloc4/CM4.6-7.pdf`

HTML : exemple

```
<ul>
  <li>tutorial : <a href="http://www.python.org/tut">http://www.python.org/tut</a></li>
  <li>documentation : <a href="http://www.python.org/doc">http://www.python.org/doc</a></li>
  <li>t&eacute;l&eacute;chargement de la derni&egrave;re version :
      <a href="http://www.python.org/download">http://www.python.org/download</a><br>
  </li>
 </ul>
</li>
<li>pour t&eacute;l&eacute;charger Dr Python :
   <a href="http://drpython.sourceforge.net/">http://drpython.sourceforge.net/</a>
  (vous aurez aussi besoin de la librairie graphique WxWidget :
      <a href="http://www.wxwidgets.org/">http://www.wxwidgets.org/</a>).</li>
<li>quelques transparents (PPT) sur les<a href="http://cmsdoc.cern.ch/%7Egrandic/pythonintro.ppt">
 structures de donn&eacute;es</a> Python, par Claudio Grandi (Universit&eacute;
 de Bologne)</li>
<li>une <a href="http://www.cis.upenn.edu/~cse391/cse391_2004/PythonIntro1.ppt">introduction</a>
 (PPT) aux structures de base de Python, aux instructions de base, et &agrave;
 la syntaxe, par Matt Huenerfauth (Universit&eacute; de Pennsylvanie)</li>
</ul>
```

lien



Une première remarque : URLs et URIs

- Une URL indique
  - une ressource
  - sur une machine
  - accessible par un protocole
- Généralisation
  - URI (Uniform Ressource Identifier)
  - Identifier une ressource
    - disponible sur internet : URL
    - simplement en lui donnant un nom (URName)
      - urn:ietf:rfc:2396
      - http://yannick.prie.org/mescollegues/Lionel.Medini

Une deuxième remarque : sur la normalisation

- Norme industrielle
  - Référentiel publié par un organisme officiel (ISO, AFNOR…).
  - En anglais : *standard*
- Standard
  - Référentiel publié par une entité privée
  - Si diffusion large : standard *de fait*
- Consortium
  - Ensemble d'entreprises, de centres de recherche, de particuliers qui s'allient pour définir des normes et standards sur tout et n'importe quoi
  - Gain : fournir les outils au moment où le référentiel est publié
    - JPEG (Joint Picture Expert Group) → norme ISO
    - MPEG (Moving Picture Expert Group) → norme ISO
    - W3C (World Wide Web Consortium) → standards
    - …

Pourquoi XML ?

- Objectif
  - représenter et échanger des données et des documents sur le web
- SGML
  - un peu vieux
  - trop complexe
- HTML
  - trop basique
    - document = en-tête + corps
  - mélange logique / présentation
    - balise b = bold (mise en gras) :
      `<bold>Attention !</b>`
    - bonne approche
      - `<important>Attention !</important>`
      - présenter la chaîne de caractères importante avec une mise en forme particulière (italique, rouge, gras, *etc.*)

Objectifs XML

- XML doit être facilement utilisable sur le Web
- XML doit supporter une grande variété d'applications
- XML doit être compatible avec SGML
- Il doit être facile d'écrire des programmes qui traitent des documents XML
- Le nombre d'options doit être réduit au minimum, idéalement à zéro
- Les documents XML doivent être lisibles et raisonnablement clairs
- La conception de XML doit être menée rapidement
- La description de XML doit être formelle et concise
- Les documents XML doivent être faciles à créer
- La concision du balisage XML est d'une importance minime

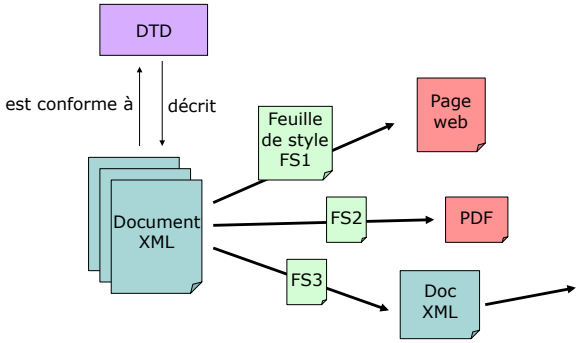
XML = métalangage

- Permet de décrire des types de documents
  - avec des DTD, des Schémas XML
- Permet de définir des instances
  - documents XML
    - répondant à un type de document
      - classique *cf.* SGML
    - simplement bien construits
      - nouveau
- Les instances peuvent décrire
  - des documents (texte balisé)
    - classique, *cf.* SGML
  - des données structurées quelconques
    - nouveau !

Principe général XML

- DTDs, Schéma
  - comment décrire les données et les documents ?
- Documents XML
  - les données et les documents eux-mêmes, dans des fichiers
- Feuilles de style
  - manière de présenter les données et les documents
- Remarque
  - on ne sait plus trop bien où sont les données, et où sont les documents !

Schéma récapitulatif

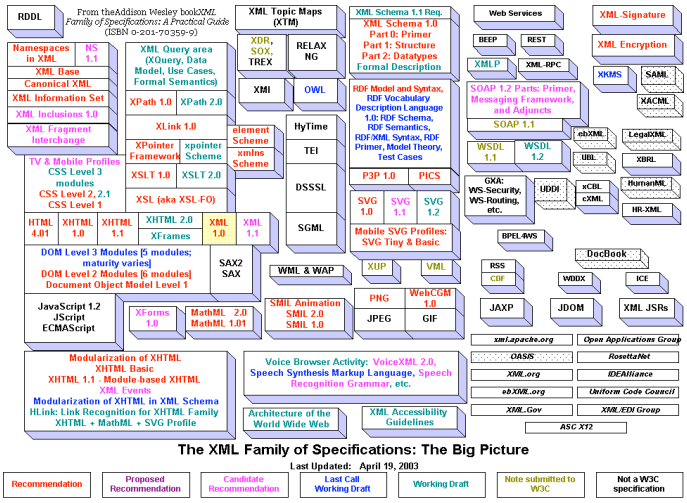


Troisième remarque : Intégration de XML dans les SI

- Stockage de données
  - simples fichiers (ex. configuration)
  - bases de données semi-structurées (requêtes, etc.)
  - bases de données documentaires
    - documents XML
    - documents XHTML (web)
- Echange de données
  - d'une base de données vers une autre (format d'échange)
  - serveur vers un navigateur : données + feuille de style
- Remarque :
  - circulation de flux XML sur un réseau :
    - utilisation de l'arbre entier (le document)
    - utilisation à la volée pour les très gros documents (exemple : BiM)

Différents langages plus ou moins standards liés à XML

- DTD / Schémas pour décrire
  - données
  - documents
- Normalisation à différents niveaux
  - W3C
  - ISO
  - organismes liés à un domaine
  - ...



Et d'autres encore !

- Suite du module :
  - XML
    - 3 CM / 2 TP
  - XPATH, XSL
    - 2 CM / 2 TP
  - (X)HTML / CSS
    - 2 CM / 2 TP
  - Projet
- Tutorat